



Universität Paderborn

Seminararbeit

Dynamische, interaktive und sensitive Karten basierend auf HTML

Konzepte zur Implementierung eines Web-Mapping-Clients

Prof. Dr. Ludwig Nastansky

Betreuer

Dipl.-Inform. Stefan Smolnik

Dipl.-Wirt.-Inf. Bernd Hesse

vorgelegt von

Jan Woltering

Matrikelnummer 6120675

Studiengang Wirtschaftsinformatik

Inhaltsverzeichnis

Inhaltsverzeichnis.....	II
Abbildungsverzeichnis.....	II
Abkürzungsverzeichnis.....	III
Angaben zur Notation.....	III
1 Einleitung.....	1
1.1 Szenario.....	1
1.2 Zielsetzung.....	2
1.3 Aufbau der Arbeit.....	2
2 Entwicklung(en) des Web-Mappings.....	3
2.1 Dynamische Karten.....	4
2.2 Interaktive Karten.....	5
2.3 Sensitive Karten und Sekundärdatenexploration.....	7
3 Anforderungen an einen Web-Mapping-Client.....	9
3.1 Dynamik, Interaktivität und Sensitivität.....	9
3.2 Sicherheit und Barrierefreiheit.....	10
3.3 Kartengestaltung, Darstellungsqualität und Datenformate.....	11
3.4 Performance.....	13
3.5 Caching und Aktualität.....	14
4 Realisierungskonzepte.....	15
4.1 Transformation der graphischen Objekte.....	15
4.2 Servergetriebene HTML-Applikation.....	16
4.3 Clientgetriebene DHTML-Applikation.....	18
5 Fazit.....	19
6 Literaturverzeichnis.....	20
Eidesstattliche Erklärung.....	23
Anhang A – Hinweise zur Implementierung.....	24

Abbildungsverzeichnis

Abbildung 1: Antialiasing.....	12
--------------------------------	----

Abkürzungsverzeichnis

BSI	Bundesamt für Sicherheit in der Informationstechnik
CSS	Cascading Style Sheets
DOM	Document Object Model
DHTML	Dynamic HTML
GIF	Graphics Interchange Format
GIS	Geographische Informationssysteme
IE	Internet Explorer
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JPEG	Joint Photographic Experts Group
JRE	Java Runtime Environment
MNG	Multiple-image Network Graphics
OGC	OpenGIS Consortium
PARC	Xerox's Palo Alto Research Center
PNG	Portable Network Graphics
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
W3C	World Wide Web Consortiums
WMS	Web Map Service
WWW	World Wide Web

Angaben zur Notation

Sämtliche *Schlüsselwörter* werden in dieser Seminararbeit bei erstmaliger Nennung kursiv hervorgehoben. Wörter mit direktem Bezug auf den Quellcode, vornehmlich Klassennamen und Methoden sowie XML-Elemente und -Attribute bzw. deren Werte, werden in einer separaten, serifenlosen `Schriftart` dargestellt.

Zur Steigerung des Leseflusses werden alle Anglizismen, mit Ausnahme der erstmaligen Nennung der Schlüsselwörter, gemäß den Regeln der deutschen Sprache dekliniert. Bei Akronymen wird auf die Deklination aufgrund der inhärenten Eigenschaften einer Abkürzung verzichtet. Vom Aspekt der Deklination abgesehen werden die Schlüsselwörter in ihrer ursprünglichen Notation angeführt.

1 Einleitung

1.1 Szenario

Karten ermöglichen wie kaum ein anderes Medium die visuelle Exploration von räumlichen oder topologischen Informationen und Gegebenheiten. Erste landkartenähnliche Wandmalereien haben ein Alter von über 8000 Jahren. Bereits aus der Antike existieren Karten auf der Basis geographischer Vermessung. Im 19. Jahrhundert erlebte die Kartografie eine Blütezeit. Aus dieser Zeit stammt eine Vielzahl der auch heute noch verwendeten graphischen Metaphern.¹

Die digitale Revolution des 20. Jahrhunderts veränderte auch das Medium Karte maßgeblich. Die fortschreitende Digitalisierung beziehungsweise Vektorisierung der räumlichen Informationen ermöglichte eine vollkommen neue Art der Interaktion mit diesem Medium. So können Karten über die reine visuelle Exploration der räumlichen und topologischen Sachverhalte hinaus zur Navigation durch einen weiter gefassten Informationsraum bestehend aus primären räumlichen Informationen und sekundären Sachinformationen dienen.

Große Potentiale bietet in diesem Zusammenhang die Einbettung von Karten in die Hypertext-Struktur des *World Wide Web (WWW)*. Durch die Bindung von Verknüpfungen an räumliche Objekte sind Web-Karten nicht mehr nur Blätter im Informationsbaum des WWW, sondern werden zu aktiven Knoten innerhalb dieses Baums. Karten werden somit ein integraler Bestandteil des Informationsraums WWW und können Funktionen übernehmen, die nicht auf die reine Vermittlung von räumlichen oder topologischen Sachverhalten beschränkt sind.

¹ vgl. Dässler (2002)

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung von grundlegenden Konzepten zur Realisierung von dynamischen, interaktiven und vor allem sensitiven Web-Mapping-Clients. Der Fokus liegt dabei auf der Einbettung der Clients in die Hypertext-Struktur des WWW. Dies bedeutet einerseits, dass ein solcher Client sich nahtlos in die Sprache des WWW (HTML) einfügt, bestenfalls selbst daraus besteht und somit das höchste Maß an Barrierefreiheit bietet, und andererseits, dass der Client die Weiterverfolgung der an die räumlichen Informationen gebundenen Verknüpfungen ermöglicht.

1.3 Aufbau der Arbeit

Analog zur Verbreitung und technologischen Entwicklung des WWW verlief die der Repräsentation von Karten in diesem Netz. In den letzten Jahren und Jahrzehnten haben sich eine Vielzahl von Methoden und Techniken zur Darstellung von Karten im WWW herausgebildet. Mit dieser Entwicklung nahm auch die Begriffsvielfalt zu. Im anschließenden Kapitel wird dieser Prozess nachgezeichnet und kategorisiert. Das zweite Kapitel dient der Begriffsklärung und Abgrenzung dieser Arbeit.

Darauf aufbauend werden im dritten Kapitel die generellen Anforderungen an einen Web-Mapping-Client umrissen. Diese umfassen nicht nur die Ausprägungen der Dynamik, Interaktivität und Sensitivität des Clients, sondern auch sekundäre Gesichtspunkte wie die Art der Darstellung, der Performanz und dem barrierefreien Zugang. Eingehende Betrachtung finden in diesem Kontext auch die wechselseitigen Abhängigkeiten der aufgestellten Anforderungen.

Zu Beginn des vierten Kapitels werden allgemeine Überlegungen zur Transformation von graphischen Objekten in HTML-konforme Repräsentationen dargelegt. Anschließend werden im Rahmen dieses Kapitels zwei Konzepte zur Realisierung eines Web-Mapping-Clients entworfen, deren konzeptueller Unterschied vornehmlich in der Gewichtung der im dritten Kapitel aufgestellten Anforderungen begründet liegt.

Abgeschlossen wird diese Seminararbeit durch Rekapitulation der aufgestellten Konzepte.

2 Entwicklung(en) des Web-Mappings

Bei den ersten im WWW bereitgestellten Karten handelte es sich um statische Rasterbilder, die durch die optische Abtastung der originalen Papierkarten erstellt wurden. Schnell wurden auf dieser Basis umfangreiche Kartensammlungen aufgebaut. Die einfache Technik und der direkte Zugriff mit Hilfe eines Web-Browsers führten dazu, dass diese Art von Web-Karten auch heute noch zu den weitest verbreiteten gehört² – auch wenn die Kartenbasis mittlerweile direkt digital erzeugt.

Zu einer der ersten Kartensammlungen dieser Art gehört die *Perry-Castañeda Library Map Collection* der *University of Texas*³. Aber nicht nur im universitären Kontext fand diese Art von Karten Verbreitung; auch Unternehmen nutzten das Präsentationspotential von Karten im WWW und fügten Filialpläne und Anfahrtsskizzen zur Kundeninformation in ihre Internet-Auftritte ein. Ebenso verwendeten Behörden und Verwaltungen dieses Verfahren früh für ihre Öffentlichkeitsarbeit.

Terminologisch fällt die Abgrenzung von statischen, dynamischen und interaktiven Kartendarstellungen durch das breite Spektrum der mittlerweile verfügbaren Technologien nicht leicht. Erschwert wird diese Abgrenzung durch die möglichen Betrachtungsebenen.

Seitens des Kartenbenutzers kann eine Web-Karte durch die Verwendung von HTML-erweiternden Datenformaten oder durch browsererweiternde Applikationen (Plug-Ins oder Applets) ein hohes Maß an Interaktivität aufweisen. Solche Karten können dem Benutzer beispielsweise die Änderung des darzustellenden Kartenausschnitts (*zoom* und *pan*) und Änderung der darzustellenden Ebenen der Karte ermöglichen. Im Bezug auf die übertragenen Informationen gesehen kann es sich bei solchen Karten technisch allerdings um eine einzelne statische auf dem Web-Server abgelegte Datei handeln.

Demgegenüber sind Web-Angebote denkbar, die dem Benutzer aktuelle Karten von hoch dynamischen Wetter- oder Verkehrsdaten in Form von einfachen Rasterbildern ohne jede Interaktionsmöglichkeit zur Verfügung stellen.

Weiterhin existieren Web-Mapping-Systeme, bei denen der Benutzer interaktiv den Kartenausschnitt, die darin enthaltenen Informationen und die Darstellungsform frei abfragen kann und das Ergebnis auf Basis der Parameter, also dynamisch, berechnet

² vgl. Dickmann (2001)

³ <http://www.lib.utexas.edu/maps/>

wird. Dementsprechend wird nur ein Ausschnitt des auf dem Server abgelegten Datenbestandes übermittelt.

Aus den obigen Szenarien ergeben die folgenden Überlegungen zur Kategorisierung der behandelten Begrifflichkeiten:

- Das *dynamische* Element einer Karte bezieht sich auf die generische Erzeugung dieser Karte; entweder aufgrund einer sich zeitlich verändernden Datenbasis oder aufgrund der Extraktion eines Teils der Datenbasis oder aufgrund von spezifischen Darstellungsanforderungen. Im Kontext des Web-Mappings ist die Dynamik einer Karte serverseitig bedingt.
- Das *interaktive* Element einer Karte wird in diesem Kontext ausschließlich clientseitig bestimmt. Die Client-Applikation ist die Schnittstelle zum Betrachter der Karte und stellt ihm die jeweiligen Kartenfunktionalitäten zur Verfügung. Ob die Karte einmalig erzeugt oder dynamisch generiert wird ist dabei unerheblich.
- *Statisch* können Karten sowohl im Hinblick auf die serverseitigen als auch die clientseitigen Gegebenheiten sein.

2.1 Dynamische Karten

Dynamische Karten zeichnen sich durch generische, serverseitige Erzeugung aus. Als Datenbasis dienen in den meisten Fällen (geographische) Datenbanken, aus denen abhängig von den Anfrageparametern die jeweiligen Informationen selektiert und in eine graphische Repräsentation transformiert werden. Bereits „*im Jahre 1993 wurde vom Xerox's Palo Alto Research Center (PARC) in Kalifornien (USA) ein erstes kartengestütztes Online-System im Internet zur Verfügung gestellt.*“⁴ Der so genannte Xerox PARC Map Viewer war in der Lage, abhängig von per HTTP übermittelten Benutzerangaben Karten auf der Basis öffentlich zugänglicher Daten zu erzeugen. Im Laufe der Zeit entwickelten sich zahlreiche Dienste, die in Kombination mit entsprechenden Client-Applikationen dem Benutzer eine wesentlich intuitivere Exploration der zum Teil mehrere Gigabyte großen geographischen Datenbestände ermöglichten.

Das Wort „Viewer“ wird heutzutage einer eigenständigen, clientseitigen Applikation zugeordnet. Serverseitig wird demgegenüber eher von „Diensten“ gesprochen. Im

⁴ Harder (1998)

Bereich der Kartenerzeugung prägt demnach folgerichtig die Bezeichnung *Web Map Service (WMS)* den Sprachgebrauch.

Unter diesem Namen hat auch das *OpenGIS Consortium (OGC)* im Jahr 1999 einen ersten Versuch zur Standardisierung der HTTP-basierten, serverseitigen Kartenerzeugung vorgenommen. Diese WMS-Spezifikation⁵ behandelt dabei nicht die Funktionsweise eines solchen Dienstes oder gibt bestimmte Ausgabeformate vor, sondern beschreibt ausschließlich, mit welchen Operationen Karten, Sekundärdaten und Metadaten des Dienstes angefragt werden können.

2.2 Interaktive Karten

Die Interaktivität einer Karte wird primär durch die Client-Anwendung bestimmt. Als grundlegende Funktionalitäten einer interaktiven Karte gelten die Möglichkeiten zur Maßstabsänderung und zum Schwenken. Zum weitergefassten Funktionsumfang gehören eine zusätzliche Übersichtskarte mit Navigationsfunktion und die Auswählbarkeit der darzustellenden Kartenebenen. Darüber hinaus können die Client-Applikationen dem Nutzer Suchfunktionen anbieten oder ihn zur Sekundärdatenexploration befähigen. Bei speziellen geographischen Applikationen kann zudem auch die Wahl des Koordinaten bzw. Projektionssystems möglich sein.⁶

Die Client-Applikationen können durch eine Vielzahl von verschiedenen Technologien implementiert werden. Dazu gehören auf der einen Seite rein browserbasierte Techniken auf der Basis von HTML und je nach System Javascript, auf der anderen Seite browsererweiternde Applikationen auf der Basis von Plug-Ins oder Applets.

Zu den wichtigsten Kriterien zur Kategorisierung der auf dem Markt vertretenen Softwarelösungen zur Kartendarstellung im WWW zählen die Granulierung und die Art der übertragenen Daten. Maßgeblich für die Unterscheidung ist dabei, ob der Client die Rendering der graphischen Elemente einer Karte in Bezug auf die geforderte Auflösung, den gewünschten Kartenausschnitt und den Maßstab eigenständig übernimmt, oder ob der Client lediglich die Anzeige der vom Server als Rasterbild übermittelten Karte vornimmt.

Klassische Web-Browser unterstützen von Haus aus lediglich die Rasterformate *JPEG*, *GIF* und in den aktuellen Versionen auch *PNG*. Andere Grafikformate, insbesondere

⁵ OGC (2001)

⁶ siehe beispielsweise http://web2.cast.uark.edu/local/mapper/make_map.html

Vektordatenformate, werden ohne Erweiterung durch Plug-Ins nicht oder nur herstellerproprietär unterstützt. Folglich führt bei einem reinen Browser- und somit HTML-basierten Client jegliche Interaktion mit der Karte wie zoom oder pan zu einer weiteren Abfrage der jeweiligen Karte und deren Transport über das Netz. Durch die Verwendung von transparenzunterstützenden Rasterformaten wie GIF oder PNG kann allerdings eine feinere Granulierung der zu übertragenden Daten vorgenommen werden, da die Karte so aus mehreren sich überlagernden Bildern zusammengesetzt werden kann. Weiterhin ist bei einer Verkleinerung des Maßstabs bei ansonsten identischem Informationsgehalt der Karte die Möglichkeit zur browserseitigen Interpolation gegeben. In der Regel greifen die HTML-Web-Mapping-Clients auf Map Server zurück, die auf Basis der jeweiligen Anfrageparameter die Karte dynamisch erzeugen. Allerdings existieren auch Erweiterungen für Desktop-GIS-Applikationen⁷, die die gewünschten Daten in Kartenausschnitte verschiedener Maßstäbe und in die zur Navigation benötigten HTML-Dokumente transformieren. Die Bilder und HTML-Dokumente können dann als statische „Web-Applikation“ auf einem einfachen Web-Server ohne jede Erweiterung abgelegt werden.

Durch die Verwendung von Plug-Ins ist auch die Einbettung von Vektorgrafiken in HTML-Seiten möglich – was eine clientseitige Rendering impliziert. Zu den im Web gebräuchlichsten Vertretern dieser Art gehören in der 2D-Welt das vom *WWW Consortium (W3C)* spezifizierte *Scalable Vector Graphics (SVG)*-Format und das proprietäre Format *Flash* der Firma MacroMedia. Bei allen Plug-Ins für diese Formate sind bereits zoom- und pan-Funktionen integriert. Zudem sind beide Formate (SVG ist auch als offene Variante von Flash gedacht) scriptfähig, so dass sich auf eine relativ einfache Art und Weise entweder mittels diese Formate verarbeitender Grafikprogramme oder durch spezielle Erweiterungen für Desktop-GIS-Applikationen Karten mit großer Interaktionsvielfalt erzeugen lassen. Bei dieser Art von Karten ist aber die übertragene Informationsmenge konstant; sämtliche Kartendaten werden zu Beginn in Form einer Datei übertragen. Neuere Entwicklungen in diesem Bereich versuchen die Vorzüge der clientseitigen Rendering mit der Variabilität und Dynamik eines die Vektorformate generisch erzeugenden Map Servers zu verbinden.

Neben den Plug-Ins für die genannten Vektorformate bieten alle namhaften GIS-Hersteller für ihre proprietären Karten- oder Datenformate entweder Plug-Ins oder eigenständige, direkt vom Browser gestartete Viewer an. Der Funktionsumfang dieser

⁷ z.B.: Image-Mapper der Firma alta4 siehe: http://www.alta4.com/eng/products_e/im/index.html

Programme erreicht dabei schon beinahe den von Desktop-GIS-Applikationen. So kann der Benutzer beispielsweise auch eigene Karten zusammenstellen.⁸

Um das Manko der Plattformabhängigkeit und Inkompatibilität bei den Plug-Ins seitens der Browser und bei den Viewern seitens der Betriebssysteme zu umgehen, werden auch Lösungen auf Basis von Java eingesetzt. Grundsätzlich hat eine solche Entwicklung von den Performance-Verlusten abgesehen, die durch die Interpretation des Bytecodes und die mangelnde Systemnähe verursacht werden, das gleiche Potential wie das native Plug-In-Pendant. Ein herausragendes Beispiel für dynamische und interaktive Karten unter Verwendung von Java bilden die Entwicklungen der Firma *map24*⁹.

2.3 Sensitive Karten und Sekundärdatenexploration

Mechanismen zur Exploration von Sekundärdaten basieren zwangsläufig auf einer diskreten Ausprägung der räumlichen Primärdaten. Die Ermittlung der räumlichen Informationsobjekte einer Karte, die mit einer bestimmten Bildkoordinate korrespondieren, kann sowohl client- als auch serverseitig erfolgen.

Das OpenGIS Consortium hat eigens für diese Funktionalität eine optionale Operation in die WMS-Spezifikation aufgenommen. Allerdings erfordert die serverseitige Ermittlung einerseits einen hohen Kommunikationsaufwand, da jeder Aufruf dieser Funktion zu einer neuen Abfrage des Servers führt, und andererseits einen hohen Speicher- oder Rechenaufwand, da die jeweilige Repräsentation einer Karte bei jeder Abfrage neu berechnet oder in einem Cache vorgehalten werden muss. Neben der evtl. hohen Wartezeit zur Beantwortung der Anfrage hat die Verwendung der serverseitigen Technik für den Benutzer weitere Nachteile:

- Vor der Übermittlung der Anfrage ist nicht sichergestellt, ob der Anwender überhaupt Bereiche der Karte anfragt, denen Sekundärdaten zugeordnet sind. Der Anwender erhält so gegebenenfalls auch leere Antworten.
- Sekundärdaten können ausschließlich explizit angefragt werden. Die automatische Einblendung von kurzen Beschreibungen oder andere sensitive Reaktionen beim Überfahren eines Gebietes mit dem Cursor sind nicht möglich.

⁸ siehe: <http://www.esri.com> und <http://www.mapguide.com>

⁹ siehe: <http://www.map24.de>

Kann der Client das Mapping der Bildschirmkoordinaten zu den räumlichen Informationsobjekten der Karte übernehmen, können die obigen Nachteile umgangen werden. Neben der Fähigkeit der Client-Applikation die nötigen Vektordaten verarbeiten zu können, ist für den Aufbau von sensitiven Karten die Bedienung des Endgerätes mit einem Zeigesystem notwendig. Als Beispiel für zeigerlose Eingabegeräte sind vornehmlich Touchscreen-Systeme, die häufig im mobilen Bereich Verwendung finden, zu nennen. Nichtsdestotrotz bleiben auch in diesem Fall die anderen Vorteile des clientseitigen Mappings bestehen.

Sowohl das HTML-Format als auch die beiden Vektorformate Flash und SVG können die Vektordaten verarbeiten und Ereignisse beim Überfahren mit dem Cursor auslösen. Allerdings wird bei allen Formaten bei sich überlappenden Flächen lediglich auf die oberste reagiert. Dieses Verhalten wird von den proprietären Entwicklungen der GIS-Hersteller zumindest beim Anklicken vermieden.

3 Anforderungen an einen Web-Mapping-Client

3.1 Dynamik, Interaktivität und Sensitivität

Ziel bei der Entwicklung von Web-Mapping-Lösungen sollten Systeme sein, die dem Benutzer einen interaktiven, performanten und intuitiven Zugriff auf einen möglichst großen Bestand von räumlichen und sachlichen Daten gewährleisten. Insbesondere Systeme mit clientseitiger Rendering und dynamischen Kartenservern bieten ein Höchstmaß an Performanz, Darstellungsqualität und Benutzerkomfort und unter Verwendung von scriptfähigen Vektordatenformaten auch dem Kartenautor mannigfaltige Erweiterungsmöglichkeiten hinsichtlich der Interaktivität und Sensitivität.¹⁰

Jedoch verursacht der zur Darstellung der Vektordaten notwendige Einsatz von browsererweiternden Technologien wie Plug-Ins oder Applets besonders im Hinblick auf sicherheitsrelevante Aspekte, Verfügbarkeit, Plattformunabhängigkeit und Barrierefreiheit vielfältige Probleme und können die potentielle Nutzergruppe stark einschränken:

- Die Installation von Plug-Ins und die für Applets notwendige *Java Runtime Environment (JRE)* erfordern bei den meisten Betriebssystemen administrative Berechtigungen.
- Plug-Ins werden direkt für spezielle Betriebssysteme und/oder Browser entwickelt. Die Verfügbarkeit oder der volle Funktionsumfang der Plug-Ins ist selbst für die drei am weitesten verbreiteten Kombinationen *Microsoft Windows XP - Internet Explorer (IE)*, *Mac OS X – Safari* und *Linux – Firefox* nicht gewährleistet. Bei der momentanen Vielfalt von Browsern und Betriebssystemen bzw. deren immer noch eingesetzten, nicht mehr aktuellen Versionen und der mangelnden Standardisierung lassen sich keine plattformunabhängigen Lösungen auf der Basis von Plug-Ins realisieren.
- Der Einsatz von Plug-Ins oder Applets verlangt eingeschränkte Sicherheitsrichtlinien seitens des Browsers. Deren Veränderung kann das Sicherheitsniveau des Computers stark beeinträchtigen und verlangt zudem meist auch administrative Berechtigungen.

¹⁰ vgl. Woltering (2004)

- Mit Ausnahme des offenen SVG-Standards sind sämtliche bisher angesprochenen Formate herstellerproprietär und demnach meist schlecht dokumentiert, was Dritten die Aufbereitung der Karten für andere Kontexte wie zum Beispiel für sehbehinderte Menschen stark erschwert.

Ausschließlich auf HTML-basierende Mapping-Clients umgehen die obigen Probleme weitgehend. Allerdings stellt die Implementierung eines dynamischen, interaktiven und sensitiven HTML-Systems wesentlich höhere Anforderungen an den Map Server, die verwendete Hardware und das Netzwerk. Der verwendete WMS muss in der Lage sein die Karten in den geforderten Rasterformaten zu generieren, was im Vergleich zur bloßen Umwandlung von geographischen Informationen in graphische Elemente wesentlich mehr Ressourcen erfordert. Zudem ist zur Gestaltung von sensitiven Karten unter HTML die dynamische Erzeugung einer so genannten *image map* zwingend erforderlich. Weiterhin führt jede Interaktion mit der Karte auch bei einer konstanten oder teilweise modifizierten Informationsmenge evtl. zu einer kompletten Neuabfrage. Darüber hinaus stehen die im Folgenden näher dargelegten Anforderungen und konzeptionellen Möglichkeiten in einer starken Wechselbeziehung, so dass die höhere Gewichtung einer bestimmten Anforderung meist in Konflikt zu einer anderen steht.

3.2 Sicherheit und Barrierefreiheit

Ein hohes Maß an Interaktivität kann auch bei HTML-Clients durch den konsequenten Einsatz von Javascript und dem vom W3C spezifizierten *Document Objects Model (DOM)* direkt clientseitig umgesetzt werden.¹¹ So ist beispielsweise die (De-)Aktivierung von einzelnen Ebenen einer Karte – sofern sie sich aus sich überlagernden, transparenten Bildern zusammensetzt – durch einfache Modifikation des `style`-Attributes möglich. Aber auch schon die Verwendung von Javascript führt zu sicherheitstechnischen Bedenken:

„Die Funktion [Anm. d. Verf.: einer Internet-Seite] im Zusammenhang mit DHTML ist nur gewährleistet, wenn die Ausführung von Javascript im Browser eingeschaltet wird. Dadurch wird aber das Sicherheitsniveau des Browsers herabgesetzt. Das BSI [Anm. d. Verf.: Bundesamt für Sicherheit in der Informationstechnik] lehnt deshalb die Verwendung von dynamischen Inhalten aus Sicherheitsgründen strikt ab.“¹²

¹¹ Bei der Kombination von HTML und Javascript wird auch von *Dynamic HTML (DHTML)* gesprochen

¹² BSI (2003) S. 74

Sollte dennoch Javascript zur Verbesserung des Bedienkomforts eingesetzt werden, müssen auf der einen Seite „*geräteabhängige Event-Handler*“¹³ (..) *strikt vermieden werden, denn sie sind nur von Benutzern auszulösen, die dieses Gerät besitzen und bedienen können*“¹⁴ und auf der anderen Seite „*müssen die Internet-Seiten auch dann verwendbar sein, wenn Scripts (...) deaktiviert sind*“¹⁵. Weiterhin ist zwar mittlerweile vom W3C das DOM und der Zugriff darauf spezifiziert worden, wird aber insbesondere von älteren Browsern nicht oder nur teilweise unterstützt, wodurch weitere Barrieren bei der Nutzung von DHTML-basierten Web-Applikationen aufgebaut werden.

Der Verzicht auf die Verwendung von DHTML führt allerdings zur kompletten Verlagerung der Applikationslogik auf den Server. Jegliche Interaktion mit der Client-Applikation führt zu einer Anfrage des Servers. Der Status der Client-Applikation muss für jeden Benutzer über die Anfrageparameter und je nach Komplexität zusätzlich über ein ressourcenbelastendes Sessionmanagement erfolgen. Außerdem lassen sich bestimmte Interaktionsmöglichkeiten wie wahlfreies Schwenken oder Zoomen nicht oder nur äußerst aufwändig implementieren.

3.3 Kartengestaltung, Darstellungsqualität und Datenformate

Die Bildschirmvisualisierung von diskreten räumlichen Daten führt zwangsläufig zu einer Rasterung. Der damit einhergehende Qualitätsverlust kann durch Bildverbesserungstechniken wie *Antialiasing*, also die Abmilderung der durch die Rasterung entstandenen „Treppeneffekte“, teilweise ausgeglichen werden.¹⁶ Durch *Antialiasing* kann die Lesbarkeit - insbesondere von Texten - deutlich gesteigert werden, allerdings geht die Abstufung der Kontraste mit einer gewissen Gesamtunschärfe einher. Der optische Eindruck einer so überarbeiteten Karte wird weicher und somit meist als gefälliger gewertet. Da browserseitiges *Antialiasing* vom W3C als Gestaltungseigenschaft nicht spezifiziert wird und lediglich der Internet Explorer von Microsoft einen solchen Filtereffekt vorsieht, muss der Map Server dazu in der Lage sein.

¹³ Dazu zählen sowohl `onmouseover` und `onclick` als auch `onkeydown`

¹⁴ BSI (2003) S. 78

¹⁵ BSI (2003) S. 74

¹⁶ vgl. Dickmann 2003

Problematisch ist Antialiasing in zweierlei Hinsicht:

Durch das Einfügen von Zwischenwerten steigt der Farbraum des Bildes an. Das Ursprungsbild des abgebildeten Beispiels ließe sich mit einer zweifarbigen Farbpalette, also mit einer Bittiefe von eins, abspeichern.

Zur Abmilderung der Treppeneffekte wurden allerdings mindestens fünf weitere Farben eingefügt, so dass mindestens drei Bit zur Adressierung des Indexwertes eines Pixels benötigt werden. Allein bei diesem Beispiel

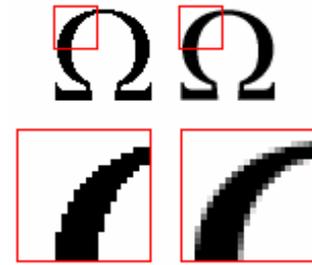


Abbildung 1: Antialiasing¹⁷

geht das Antialiasing mit minimal einer Verdreifachung¹⁸ des Datenvolumens einher. Bei mehrfarbigen Ursprungskarten kann das mögliche exponentielle Wachstum der Zwischenwerte zu einer noch wesentlich stärkeren Vervielfachung des Datenvolumens führen.

Ist der Hintergrund des Ursprungsbildes zudem transparent, wird das Antialiasing nicht nur auf die Farbwerte eines Pixels sondern auch auf dessen Transparenzwert, den so genannten α -Wert angewendet. Momentan ist allerdings nur das PNG-Format zur Handhabung der α -Werte in der Lage und dies sowohl indiziert als auch im Vollfarbmodus mit 32 Bit.

Die mangelhafte Unterstützung von α -Werten bei anderen Rasterformaten tritt auch bei der Verwendung von halbtransparenten Farben bei Linien oder Flächenfüllungen auf. Zwar können diese Werte in Bezug auf die Hintergrundfarbe des Bildes beim GIF-Format korrekt berechnet werden, aber die Bilder sind in diesen Bereichen bei der Überlagerung mehrerer Bilder trotzdem deckend, da immer nur ein Farbwert als transparent vermerkt werden kann.

Aus diesen Überlegungen lassen die folgenden Verwendungsmöglichkeiten der Bildformate JPEG, GIF, und PNG für kartographische Zwecke ableiten:

- Das PNG-Format eignet sich hervorragend für sämtliche Verwendungszwecke. Die variable Indizierung, die verlustlose Kompression und die Handhabung von bis zu 2^{48} Farben und 2^{16} Transparenzwerten erfüllen alle Anforderungen - zumindest im Bereich des Web-Mappings und unter Vernachlässigung von

¹⁷ Quelle: <http://de.wikipedia.org/wiki/Antialiasing>

¹⁸ Unter Vernachlässigung der durch Kopfdaten bedingten Fixkosten, die bei sehr kleinen Bildern mit guter Kompression signifikant ins Gewicht fallen können.

Animationen¹⁹. Lediglich die fehlende Unterstützung von älteren und die zum Teil mangelhafte Unterstützung von aktuellen Browsern sprechen gegen dieses Format.

- Das GIF-Format ist durch die Festlegung auf eine 8-bittige Farbpalette zur Darstellung von vollfarbigen Rasterdaten nur bedingt geeignet. Weiterhin kann es zwar zum Aufbau von Karten bestehend aus sich überlagernden Einzelbildern verwendet werden. Dabei muss aber seitens der Kartengestaltung auf halbtransparente Farbwerte und seitens der Darstellungsqualität auf Antialiasing verzichtet werden. Lediglich bei der Realisierung von animierten Symbolen kann auf dieses Format momentan noch nicht verzichtet werden.
- Sofern Karten immer in Gänze erzeugt werden, bietet das JPEG-Format durch die volle Farbtiefe und qualitätsabhängige Kompressionsalgorithmen ein sehr gutes Verhältnis von Darstellungsqualität und Datenvolumen.

3.4 Performance

Die Performance eines verteilten Client-Server-Systems hängt maßgeblich von der Granularität der übertragenen Daten ab, also von der Anzahl der Anfragen, dem Datenvolumen und dem Aufwand zur Erzeugung der Daten.

Der Aufwand zur Erzeugung der Karten korrespondiert hauptsächlich mit der Auflösung der Bilder, der Kompression, der Datenmenge und der Art der Rendering. Besonders das Antialiasing verlangt nochmals eine rechenintensive Analyse des gesamten Bildes und erhöht zudem das Datenvolumen durch größere Indizes und schlechtere Komprimierbarkeit.

Weiterhin führt beispielsweise der Aufbau einer Karte aus sich schichtweise überlagernden Einzelbildern zwar zu höherer clientseitiger Interaktivität, bedingt aber auch, dass jedes Bild einzeln angefragt werden muss, wobei jeder Verbindungsaufbau sowohl client- als auch serverseitig zu einem zusätzlichen Ressourcenverbrauch führt. Die einmalige Übertragung aller Ebenen in einem Bild, dessen Datenvolumen evtl. sogar die Summe der Einzelvolumen übersteigt, ist durch den geringeren Verbindungsaufbau-Overhead wesentlich schneller. Die Vorteile der feineren Datengranularität kommen erst mit deren Ausnutzung zur Geltung – ungenutzt geht sie demnach mit höheren Kosten einher.

¹⁹ Es existiert ein PNG ähnliches, bislang von keinem Browser unterstütztes Format namens *Multiple-image Network Graphics (MNG)*, das auch Animationen ermöglicht.

3.5 Caching und Aktualität

Die Granularität oder auch Strukturierung der Daten spielt aber nicht nur im Hinblick auf die clientseitige Interaktivität, sondern auch besonders im Hinblick auf die Pufferfähigkeit der Daten und somit auf die Leistungsfähigkeit des Servers eine entscheidende Rolle.

„Caching stellt eine der effektivsten Möglichkeiten dar, die Antwortzeit einer Anfrage signifikant zu verringern. Statt einer Neuberechnung der dynamischen Daten werden diese einfach aus dem Speicher gelesen.“²⁰

Sichergestellt wird ein Caching – sofern es auf dem HTTP beruht – durch eine möglichst feste Strukturierung der Anfrage-URL. Im Bereich der dynamischen Kartenerzeugung variieren die URL vornehmlich durch die Wahl der in der Karte enthaltenen Ebenen und deren Reihenfolge, sowie durch den gewünschten Kartenausschnitt.²¹

So sollte zur Eingrenzung der möglichen Kartenausschnitte eine Abfrage lediglich auf festgelegten Kacheln anstatt auf der freien Wahl der gewünschten räumlichen Ausdehnung basieren. Dabei muss die Client-Applikation so gestaltet sein, dass für jeden Benutzer die selben Kacheln abgefragt werden. Bei einer Kachelgröße von beispielsweise 200×200 Pixel wird für einen bestimmten Maßstab die Zahl der möglichen Anfragen um den Faktor 40.000 verringert.

Wird bei einer Client-Applikation die Karte nicht aus sich überlagernden Bildern zusammengesetzt, ist zu bedenken, dass bei einer Gesamtebenenanzahl n die benutzerseitige Deaktivierbarkeit jeder einzelnen Ebene zu 2^n möglichen Anfragen und die benutzerseitige Änderung der Ebenenreihenfolge zu $n!$ möglichen Anfragen führen kann.

²⁰ Woltering (2004)

²¹ vgl. auch Woltering (2004) S. 36f.

4 Realisierungskonzepte

Die Vielzahl der im obigen Kapitel aufgestellten Anforderungen an ein Web-Mapping-System bzw. deren Wechselwirkungen und die sich daraus ergebenden Konsequenzen machen die Konzeption eines Clients, der allen Anforderungen gerecht wird, unmöglich. Insofern werden im Folgenden zwei Clients konzipiert:

- Ein servergetriebener HTML-Client, der vollständig auf die Verwendung von Javascript verzichtet und zudem lediglich Methoden benutzt, die dessen Einsatz auf allen – auch älteren – Browsern und Betriebssystemen ermöglicht.
- Ein clientgetriebener DHTML-Client, der unter Verwendung von Javascript die gesamte Applikationslogik selbst ausführt und ausschließlich mit einem WMS zur Datenabfrage kommuniziert.

Als Basis für beide Clients wird von einem WMS ausgegangen, der die Karten sowohl in alle notwendigen Formate rendern als auch in eine image map transformieren kann. Darüber hinaus wird ein vom Kartenautor erstelltes Konfigurationsdokument angenommen, das für sämtliche Ebenen der Karte deren räumliche Ausdehnung, evtl. maßstabsabhängige Sichtbarkeit und mögliche Sensitivität beinhaltet.

4.1 Transformation der graphischen Objekte

Zur Realisierung einer clientseitigen image map sieht die aktuelle HTML 4-Spezifikation das so genannte `map`-Element vor. Es dient als Containerelement für die einzelnen Flächen der image map und dessen `name`-Attribute zur Verknüpfung mit dem jeweiligen Bild, über das die image map gelegt wird. Als einziges Kindelement eines `map`-Elements wird das `area`-Element unterstützt. Das `area`-Element wird von der Flächendefinition abgesehen von allen Browsern hinsichtlich der Event-Behandlung, des Titels und der Verlinkung genauso wie das `a`-Element interpretiert.

Zur Definition der Flächen stehen die drei Formen Rechteck, Kreis und Polygon zur Verfügung, wobei ein Rechteck durch die obere linke und untere rechte Ecke, ein Kreis durch den Mittelpunkt und den Radius und ein Polygon durch die Abfolge der Koordinaten des äußeren Umrings definiert wird.

Die wesentlich komplexeren geometrischen Formen, die die vektorielle Beschreibung einer Karte enthalten kann, wie (Multi-)Polygone und (Multi-)Polylinien, können nur durch Zerlegung korrekt abgebildet werden. Auch Polylinien erhalten durch ihre Strichstärke bei der Rendering eine flächige Darstellung. Die einzelnen Segmente einer

Linie können dabei jeweils durch ein rechteckiges Polygon substituiert werden. Des Weiteren können Polygone mit inneren Ausschlüssen durch Triangulation in mehrere ausschlussfreie Polygone zerlegt werden. Ein solches Verfahren ist aber sehr aufwändig und im Hinblick auf die Vernachlässigbarkeit hoher geometrischer Präzision bei der Gestaltung von sensitiven Karten gegebenenfalls unnötig. Um die semantische Integrität eines geometrischen Objektes bei der Zerlegung zu wahren, müssen aufgrund der hierarchielosen Struktur innerhalb eines `map`-Elementes den entstandenen `area`-Elementen gegebenenfalls Attribute redundant angefügt werden.

Symbole in einer Karte können entweder relativ aufwändig analog zu Polygonen verarbeitet oder auf die Grundformen Kreis und Rechteck reduziert werden – besonders wenn es sich bei den Symbolen um externe Rasterbilder handelt.

4.2 Servergetriebene HTML-Applikation

Die oberste Priorität bei der Konzeption dieses Web-Mapping-Clients liegt auf dem vollständigen Verzicht auf Javascript, daher muss die gesamte Applikationslogik serverseitig ausgeführt werden. Darüber hinaus wird aus Gründen der Barrierefreiheit eine geräteunabhängige Interaktion mit dem Client verlangt, also sowohl mittels physischer Geräte wie Zeigersystemen, Tastatur und Touchscreen als auch über Screenreader oder Spracherkennungssysteme. Dieses bedeutet zudem, dass jeder Link ein festes Verweisziel haben muss und somit einen Verzicht auf serverseitige image maps. Da dieser Client weiterhin von allen Browsern und von assistiven Technologien wie Screenreadern interpretiert werden können soll, werden ausschließlich die Spezifikationen HTML 4.0 und CSS 2 vom W3C als technische Basis festgelegt. Dadurch werden die verwendbaren Bildformate auf JPEG und GIF beschränkt. Da außerdem die Gestaltung und Darstellung der Karten keinen Beschränkungen unterliegen soll, verhindert die Verwendung von Techniken wie Antialiasing, vollfarbigen Rasterdaten und halbtransparenten Füllungen, Linien und Symbolen den schichtweisen Aufbau der Karten durch einzelne GIF-Bilder. Eine Karte muss daher in Gänze als JPEG-Bild gerendert werden, was weitreichende Konsequenzen hinsichtlich der Interaktivität in Bezug auf die (De-)Aktivierbarkeit einzelner Ebenen bzw. die Änderbarkeit der Ebenenreihenfolge hat. Die Implementierung dieser Funktionen würde zu einer starken Individualisierung der Kartengestaltung und somit zu einer wesentlichen schlechteren Ausnutzung eines Caches und folglich zu einer linearen Korrelation zwischen Benutzeraufkommen und Auslastung des WMS führen. Stattdessen sollte der Kartenautor die Sichtbarkeit jeder einzelnen Ebene

maßstabsabhängig konfigurieren und bei sich in der Ausprägung überdeckenden Ebenen die Sichtbarkeit der verdeckten Ebenen gegebenenfalls durch die Variation der Ebenentransparenz der verdeckenden Ebenen sicherstellen. Eine sinnvolle Implementierung eines Caches wird allerdings erst durch die gekachelte Abfrage einer Karte möglich. Neben den serverseitigen Vorteilen bietet diese Methode dem Benutzer zudem den Vorteil, dass beim Verschieben des Kartenausschnitts nur die Kacheln vom Server abgefragt werden müssen, die in einem der vormaligen Kartenausschnitte noch nicht geladen wurden. Bei der Implementierung der Navigationsfunktionalitäten sind so lediglich zwei Möglichkeiten abzuwägen:

- Die möglichen sichtbaren Kartenausschnitte werden nur durch die Struktur der Kachelung bestimmt. Das heißt, die Größe des Kartenausschnitts muss dem Vielfachen der Kachelgröße entsprechen. Ein Verschieben des Kartenausschnitts ist dabei folglich nur um ein Vielfaches der Kachelgröße möglich. Weiterhin ist bei dieser Art der Umsetzung nicht jeder Kartenmittelpunkt möglich, was vor allem bei zentrierten Maßstabsänderungen relevant ist.
- Die Größe des Kartenausschnitts ist unabhängig von der Kachelgröße. Die Kacheln werden innerhalb des Kartenausschnitts frei positioniert und gegebenenfalls abgeschnitten. Diese Technik ermöglicht freies Schwenken, geht allerdings je nach Kachel- und Kartenausschnittsgröße mit einem mehr als doppelt so hohen Datenvolumen einher.

Die Serverapplikation benötigt bei dieser Gesamtkonzeption zur Bearbeitung einer Anfrage lediglich Angaben über das zu verwendende Kartenkonfigurationsdokument, einen räumlichen Bezugspunkt, beispielsweise den der oberen linken Ecke des gewünschten Kartenausschnitts und über den gewünschten Maßstab. Auf Basis dieser Parameter und der Vorgabewerte für die Kachel- und Kartenausschnittsgröße können dann die Parameter für die nächstmöglichen Navigationsfunktionen berechnet werden. Mit zusätzlichem Bezug auf das Konfigurationsdokument können weiterhin sowohl die Kartenebenen für die Rendering als auch die möglichen Ebenen der image map ermittelt werden. Die einzelnen Kacheln können über das `img`-Element als Referenz auf den eigentlichen WMS in das Ausgabedokument eingefügt werden. Die image map muss hingegen entweder kachelweise oder über alle Kacheln der Karte serverseitig vom WMS abgefragt und in das Ausgabedokument eingebettet werden, da eine externe Verknüpfung von keinem namhaften Browser unterstützt wird.

4.3 Clientgetriebene DHTML-Applikation

Die Konzeption dieses Web-Mapping-Systems folgt dem Ziel die gesamte Applikationslogik in den Client zu integrieren, so dass sich nach dem einmaligen Übertragen des nötigen Programmcodes und der Applikationsseite jegliche weitere Kommunikation auf Anfragen an den WMS beschränken lassen. Dabei soll die Granularität der übertragenen Daten so beschaffen sein, dass bei konstantem Maßstab und gleich bleibender Auflösung alle Interaktionen mit der Karte nur die Abfrage von vormals noch nicht übertragenen Daten zur Folge haben – sowohl hinsichtlich der Rasterbilder als auch der image maps. Dieses ermöglicht auf der einen Seite die direkte, clientseitige (De-)Aktivierbarkeit einzelner Ebenen, die Änderung der Darstellungsreihenfolge und freies, intuitives Schwenken sowie eine gute Ausnutzung der server- und clientseitigen Caches. Auf der anderen Seite erfordert diese feine Datengranularität aus den genannten darstellerischen Gründen die Verwendung des PNG-Formats und die Fähigkeit der Browser zur dynamischen Modifikation des DOM bezüglich der Erweiterung der image map um weitere `area`-Elemente und des zur Strukturierung der visuellen Repräsentation benötigten Teilbaums um weitere Kartenebenenkacheln.

Die Erfordernisse an die Browser hinsichtlich des PNG-Formats und die dynamische Erweiterung des DOM, die auf Grund der geforderten Sensitivität nicht nur die Rendering einer HTML-Seite sondern auch Modifikationen in Bezug auf die Behandlung von Ereignissen betrifft, schränken die Zahl der möglichen Browser stark ein. Zudem verlangen die mangelnde Konformität und das unterschiedliche Verhalten der einzelnen Browser besonders im Hinblick auf die Event-Behandlung, den Bildschirm Aufbau und die Verarbeitung des PNG-Formats gegebenenfalls browserabhängige Implementierungen.

5 Fazit

Ziel der vorliegenden Seminararbeit war die Konzeption von HTML-basierten Web-Mapping-Systemen, die nicht nur die primären Anforderungen an Dynamik, Interaktivität und Sensitivität erfüllen sondern weiteren Anforderungen in Bezug auf Barrierefreiheit und Plattformunabhängig, Darstellungsqualität und Kartengestaltung clientseitiger Interaktivität und geringer Serverbelastung genügen. Die Unvereinbarkeit der verschiedenen Anforderungen auf Grund ihrer starken wechselseitigen Abhängigkeit bzw. Beeinflussung erforderte die Konzeption zweier Varianten je nach Gewichtung der einzelnen Anforderungen. Die gewonnenen Erfahrungen aus der praktischen Umsetzung der beiden Konzepte versprechen weitreichende Einsatzmöglichkeiten der Client-Varianten.

Der HTML-Client zeichnet sich durch Barrierefreiheit, Robustheit, Einfachheit, intuitive Navigation und einen schnellen erstmaligen Seitenaufbau aus. Für viele Benutzer bedeutet der Verzicht auf die ebenenbezogenen Interaktionsmöglichkeiten und das eingeschränkte Schwenken nicht einen Makel, sondern steigert die Übersichtlichkeit und sorgt durch die Erwartungskonformität in Bezug auf andere Web-Mapping-Systeme für einen schnellen und einfachen Einstieg zur visuellen Exploration der Karten - ohne jedoch auf die Vorteile von sensitiven Karten verzichten zu müssen.

Die DHTML-Version ist in ihrem Funktions- und Interaktionsumfang durchaus mit vektordatenbasierten Systemen zu vergleichen. Lediglich bei Maßstabsänderungen, der möglichen Auflösungen und der mit dem Datenvolumen korrelierenden Performance werden die Unterschiede deutlich. Im Gegensatz zu den browsererweiternden Techniken basiert der DHTML-Client allerdings ausschließlich auf den Spezifikationen des W3C und sämtliche verwendeten Konzepte und Methoden werden bereits von den drei großen Browsern *IE*, *Firefox* und *Opera* unterstützt. Allerdings bestehen zum Teil großen Differenzen hinsichtlich der Geschwindigkeit beim Rendern der Karten und bei der dynamischen Modifikation des DOM. Insbesondere auf der *Gecko*-Renderingengine²² basierende Browser zeigen hierbei Schwächen. Diese und die unzureichende Konformität anderer Browser wie zum Beispiel *Safari* für den Macintosh werden wohl in künftigen Versionen behoben sein, so dass auch diesem Konzept eines Web-Mapping-Systems keine Beschränkungen auferlegt sind.

²² Dazu gehören u.a. Netscape, Mozilla und Firefox.

6 Literaturverzeichnis

Anderson (2000):

Anderson, Richard; Birbeck, Mark; Kay, Michael; u.a.: *XML professionell*, MITP-Verlag, Bonn, 2000

BSI (2003):

Bundesamt für Sicherheit in der Informationstechnik (BSI): *E-Government-Handbuch – Barrierefreies E-Government*, 2003

http://www.bsi.bund.de/fachthem/egov/download/4_Barriere.pdf, am 01.02.2004

Dahinden, Neumann und Winter (2003):

Dahinden, T.; Neuman, A und Winter, A. M.:

Internet-Kartengraphik mit SVG – Werkzeuge, Techniken, Anwendungen,

In: Web.Mapping 2, Hrsg.: Asch, H. und Heerman C., Wichman, Heidelberg, 2002, Seiten. 190 – 206

Dässler (2002):

Dässler, Rolf: *Visuelle Kommunikation mit Karten*, 2002

<http://fabdp.fh-potsdam.de/daessler/paper/karten.pdf>, am 02.02.2004

de Lange (2002):

de Lange, Norbert: *Geoinformatik in Theorie und Praxis*,

Springer, Heidelberg, 2002

Dickmann (2001):

Dickmann, Frank: *Web-Mapping und Web-GIS*, Westermann, Braunschweig, 2001

Harder (1998):

Harder, C: *Serving Maps on the Internet – Geographic information on the world wide web*, Redlands, CA (USA), 1998

Java Community Process (2003):

Java Community Process: *JSR-000154 Java™ Servlet 2.4 Specification*, Sun Microsystems, 2003,

<http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>, am 02.02.2004

OpenGIS Consortium (2001):

OpenGIS Consortium: *Web Map Service*, Version 1.1.1, 2001,
<http://www.opengis.org/doc/01-068r2.pdf>, am 02.02.2004

OpenGIS Consortium (2003):

OpenGIS Consortium: *OpenGIS Reference Model*, Version 0.1.2, 2003,
<http://www.opengis.org/doc/03-040.pdf>, am 02.02.2004

The Internet Society (1999):

The Internet Society: *Hypertext Transfer Protocol*, Version 1.1, 2003,
<http://www.ietf.org/rfc/rfc2616.txt>, am 02.02.2004

W3C (1996):

W3C: *PNG (Portable Network Graphics) Specification*, Version 1.0, 1996,
<http://www.w3.org/TR/REC-png.html>, am 02.02.2004

W3C (1999):

W3C: *HTML Specification*, Version 4.01, 1999,
<http://www.w3.org/TR/html4/>, am 01.02.2005

W3C (2003) b:

W3C: *Scalable Vector Graphics (SVG) 1.1 Specification*, Version 1.1, 2003,
<http://www.w3.org/TR/SVG11>, am 02.02.2004

W3C (2003) c:

W3C: *Document Object Model (DOM) Level 3 Core Specification*,
Version 1.0, 2003, <http://www.w3.org/TR/2003/CR-DOM-Level-3-Core-20031107>,
am 02.02.2004

W3C (2004) a:

W3C: *CSS 2.1 Specification*, 2004,
<http://www.w3.org/TR/CSS21/>, am 02.02.2004

W3C (2004) b:

W3C: *SVG Implementations*, 2004,
<http://www.w3.org/Graphics/SVG/SVG-Implementations.htm8>, am 02.02.2004

Winter (2000):

Winter, Andréas. M.: *Internetkartographie mit SVG*, 2000

http://www.carto.net/andre.mw/projects/atlas/internetkarto_svg_atlas_001206.pdf,
am 02.02.2004

Woltering (2004):

Woltering, Jan: *Interaktive Kar auf Basis Geographischer Informationssysteme*,

[http://gcc.upb.de/www/WI/WI2/wi2_lit.nsf/0/fac45730ac352e39c1256eb3003c385b/\\$FILE/03181102.pdf](http://gcc.upb.de/www/WI/WI2/wi2_lit.nsf/0/fac45730ac352e39c1256eb3003c385b/$FILE/03181102.pdf)/Diplomarbeit_Jan_Woltering.pdf

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Paderborn, den 29.03.2005

.....
(Jan Woltering)

Anhang A – Hinweise zur Implementierung

Bei der Implementierung der beiden Web-Mapping-Clients wurde auf den im Rahmen der Diplomarbeit „*Interaktive Karten auf Basis Geographischer Informationssysteme*“²³ entwickelte Web Map Service der Spezifikation 1.1.1 zurück gegriffen. Allerdings waren insbesondere im Hinblick auf die Erzeugung der image maps und des gekachelten Kartenaufbaus Modifikationen und Erweiterungen am bestehenden System notwendig. So wurde für die beiden Clients jeweils eine Serialisierungskomponente zur Transformation der SVG-Karte in eine image map entwickelt, wobei sich die beiden Varianten vornehmlich in der Verarbeitung von Sekundärattributen und in der Ereignisbehandlung unterscheiden. Weitere Modifikationen ergaben sich aus der Tatsache, dass sich die räumliche Ausdehnung von punktuell referenzierten Kartenelementen wie Texten und Symbolen über mehrere Kacheln erstrecken kann und somit die Kartenelemente auch abgefragt und gerendert werden müssen wenn der Referenzierungspunkt nicht innerhalb einer Kachel liegt.

Die Server-Applikation des HTML-Clients wurde als Servlet umgesetzt, das das Kartenkonfigurationsdokument (*eXmap*) als DOM einliest und auf dessen Basis neben der Karte und der image map auch die Navigationsfunktionen, eine adaptive Übersichtskarte, einen Maßstabsbalken und eine Legende erzeugt bzw. in das Ausgabedokument einfügt.

Die clientseitige DHTML-Applikation erhält das nötige Kartenkonfigurationsdokument in Form einer vom Server erzeugten und um die nötigen Attribute erweiterten Legende. Auf dieser Basis werden die nötigen Kacheln der Rasterdarstellung und der image map je nach maßstabsabhängiger bzw. benutzerseitiger Sichtbarkeit und räumlicher Ausdehnung dynamisch vom WMS abgefragt bzw. aus dem lokalen Cache geladen. Neben den in dieser Arbeit beschriebenen Basisfunktionen wie freies Schwenken, Maßstabsänderung, Ebenen(de-)aktivierbarkeit und Änderung der Ebenenreihenfolge bietet dieser Client auch eine adaptive Übersichtskarte und eine Messfunktion. Weiterhin ist je nach Browser die direkte und clientseitige Modifikation der Ebenentransparenz möglich.

„Last but not least“ verfügen beide Clients über eine Suchfunktion.

²³ Woltering (2004)